

# THE PAINS, THE GROWTH AND THE UGLY.

THE FUGLY

# THE UGLY

Cassiano Aquino

- Tech Lead DevOps Engineer
- Zendesk
- <http://syshero.org/>
- <http://github.com/caquino>
- @syshero
- Linux user since 1995
- Chef user since 2011

Wrap-up and the presentation will be available in the next few days on my blog.



CHIEF @ ZENDESK

# CHEF @ ZENDESK

## Enterprise Chef 12

- 181 roles
- 200 cookbooks
- ~2000 nodes
- ~93,600 convergences day
- ~28,000,000 attributes



# BEGINNING PARTY

Everything works

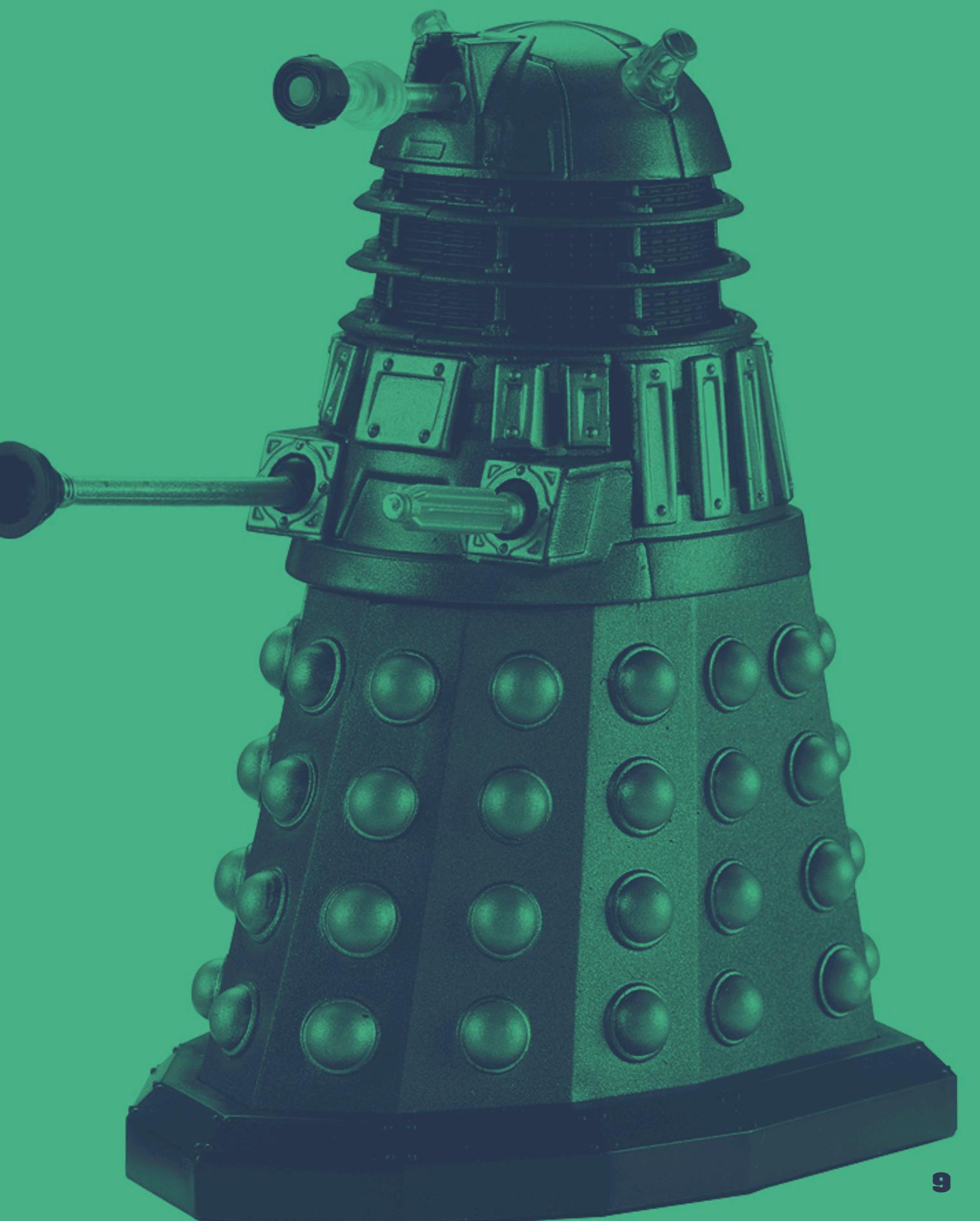
- Attributes, a lot of attributes!
- Fast convergences
- Search everywhere

# PERFORMANCE

# AFTER PARTY HANGOVER

Why it's not working anymore?

- Attribute hoarder
- Run times bigger than interval
- Search? partial search!
- DDoS
- Exterminate! Exterminate!



# DOOS?

# ALMOST!

$\text{avg(node-size)} * \text{conv/d} * \text{nodes} = \text{transferred data}$

- $\text{avg(node-size)}$ : 150KBytes
- Convergences day: 288 (1 node every 5 minutes)
- Nodes: 2000

$$150 * 288 * 2000 = 86,400,000 (\sim 82 \text{ GBytes})$$

Just a small percentage of the total traffic.

- API calls, Searches, node.save...



CAUTION



EPIC  
FAILURE

# MONITORING

Important

- Clients health 

HOW?

- Report/Exception handlers
- knife status

## 2. tmux (tmux)

```
3 days, 3:57, 1 user, load average: 14.61, 10.62, 9.6
 1 running, 1631 sleeping, 0 stopped, 0 zombie
 0.1%sy, 0.0%ni, 66.9%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%
al, 43118728k used, 22809144k free, 856696k buffers
al, 11500k used, 3891984k free, 9796664k cached
```

NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
0	7032m	2.7g	4652	S	2410	4.2	180205:22	beam.smp
0	5923m	332m	3280	S	21	0.5	25231:17	beam.smp
0	4633m	361m	4136	S	19	0.6	4173:10	beam.smp
0	40.2g	15g	12m	S	15	24.8	11853:36	java
0	255m	119m	2320	S	8	0.2	6693:33	ruby
0	184m	38m	6644	S	5	0.1	283:01.03	ruby
0	99.4m	9m	2832	S	3	0.0	19:45.66	nginx
0	182m	36m	6640	S	3	0.1	171:22.02	ruby
0	98.8m	9704	2828	S	3	0.0	19:55.54	nginx
0	99.1m	9908	2828	S	2	0.0	19:34.81	nginx
0	99.2m	9.8m	2828	S	2	0.0	19:51.43	nginx
0	99.1m	9980	2828	S	2	0.0	19:43.83	nginx
0	14.0g	41m	36m	S	2	0.1	178:17.21	postgres
0	313m	178m	2320	S	2	0.3	6987:19	ruby
0	165m	31m	1968	S	2	0.0	1890:39	ruby
0	99.2m	9.8m	2832	S	1	0.0	19:53.66	nginx
0	99.0m	9936	2828	S	1	0.0	19:43.67	nginx
0	99.0m	9888	2824	S	1	0.0	19:40.79	nginx
0	99.6m	10m	2840	S	1	0.0	19:49.80	nginx
0	125m	22m	4420	S	1	0.0	241:17.09	python
0	14.0g	39m	34m	S	1	0.1	195:04.38	postgres
0	18812	2948	1100	R	1	0.0	0:02.45	top
0	98.9m	9812	2828	S	1	0.0	19:24.07	nginx
0	99.0m	10m	2622	S	1	0.0	13:44.30	nginx

# BUT WHY EVERYTHING IS FAILING NOW?

## Important

- erchef ❤ CPU

## WHY? WHY?

- Nodes stored on postgresql
- Compressed JSON attributes
- SOLR is for searches only

# HOW DO I SPEED UP THINGS?

Old dog, new tricks

## Attributes

- Cleanup attributes
- whitelist-node-attrs

## Convergence

- Adjust convergence interval
- splay is your friend! 

## Caching

- node.run\_state



# CACHING

node.run\_state example

```
def find_lb_nodes(service_name)
  nodes = node.run_state["lb_nodes_cache"] ||=
    search(:node, "lb_service:yes")
  nodes.find{|n| n["lb_service_name"] == service_name }
end

web_nodes = find_lb_nodes("http")
db_nodes = find_lb_nodes("db")
```

# ZENDESK ARCHITECTURE

# STAGES

## Open Source

- Scale up
- Scale out

## Enterprise

- HA
- HA + Scale out
- HA + DR + Scale out



# ENTERPRISE VERSION

U.S.S. ENTERPRISE  
© ZENDESK, 2015 - CASSIANO AQUINO - CAQUINO@ZENDESK.COM  
SPACE CRUISER

MAIN  
SENSOR  
SECONDARY  
HULL

HANGAR DECK  
(SHUTTLE CRAFT)

340'

0 50 100 200  
SCALE IN FEET



# TOPOLOGY

## Configurations

- HA
- tier
- manual

# BACKENDS

```
topology "ha"
```

```
server "backend1.domain.com",
:ipaddress => "192.168.0.22",
:role => "backend",
:bootstrap => true,
:cluster_ipaddress => "194.168.1.22"
```

```
server "backend2.domain.com",
:ipaddress => "192.168.0.23",
:role => "backend",
:cluster_ipaddress => "192.168.1.23"
```

```
backend_vip "backendvip.domain.com",
:ipaddress => "192.168.0.24",
:device => "bond0"
```

# FRONTENDS

```
server "frontend1.domain.com",  
:ipaddress => "192.168.0.22",  
:role => "frontend"
```

```
server "frontend2.domain.com",  
:ipaddress => "192.168.0.23",  
:role => "frontend"
```

# APPLYING THE CONFIGURATION BACKEND AND DRBD

on backend1.domain.com

- chef-server-ctl reconfigure

manually configure DRBD on both backends

on backend1.domain.com

- chef-server-ctl reconfigure

# APPLYING THE CONFIGURATION FRONTENDS

from backend1.domain.com to all servers

- copy /etc/opscode

on the remaining servers

- chef-server-ctl reconfigure

# DR SOLUTION CHEF-SYNC

Asynchronous distribution

- cookbooks
- environments
- roles
- data bags



Open Source

# OPEN SOURCE

backend

```
postgresql['md5_auth_cidr_addresses'] = [ "127.0.0.1/32",  
                                         "::1/128",  
                                         "192.168.0.0/24" ]
```

# OPEN SOURCE

frontend disable services

```
bootstrap["enable"] = false
rabbitmq["enable"] = false
chef_solr["enable"] = false
chef_expander["enable"] = false
bookshelf["enable"] = false
chef_server_webui["enable"] = false
postgresql["enable"] = false
```

# OPEN SOURCE

use backend services

```
rabbitmq["vip"] = "192.168.0.8"
rabbitmq["node_ip_address"] = "192.168.0.8"
chef_solr["vip"] = "192.168.0.8"
bookshelf["vip"] = "192.168.0.8"
postgresql["vip"] = "192.168.0.8"
lb["upstream"] = {
    "erchef" => [ "127.0.0.1" ],
    "chef-server-webui" => [ "192.168.0.8" ],
    "bookshelf" => [ "192.168.0.8" ]
}
```

# OPEN SOURCE

small patch

/opt/chef-server/embedded/cookbooks/chef-server/  
templates/default/sv-erchef-run.erb

comment out

/opt/chef-server/bin/wait-for-rabbit



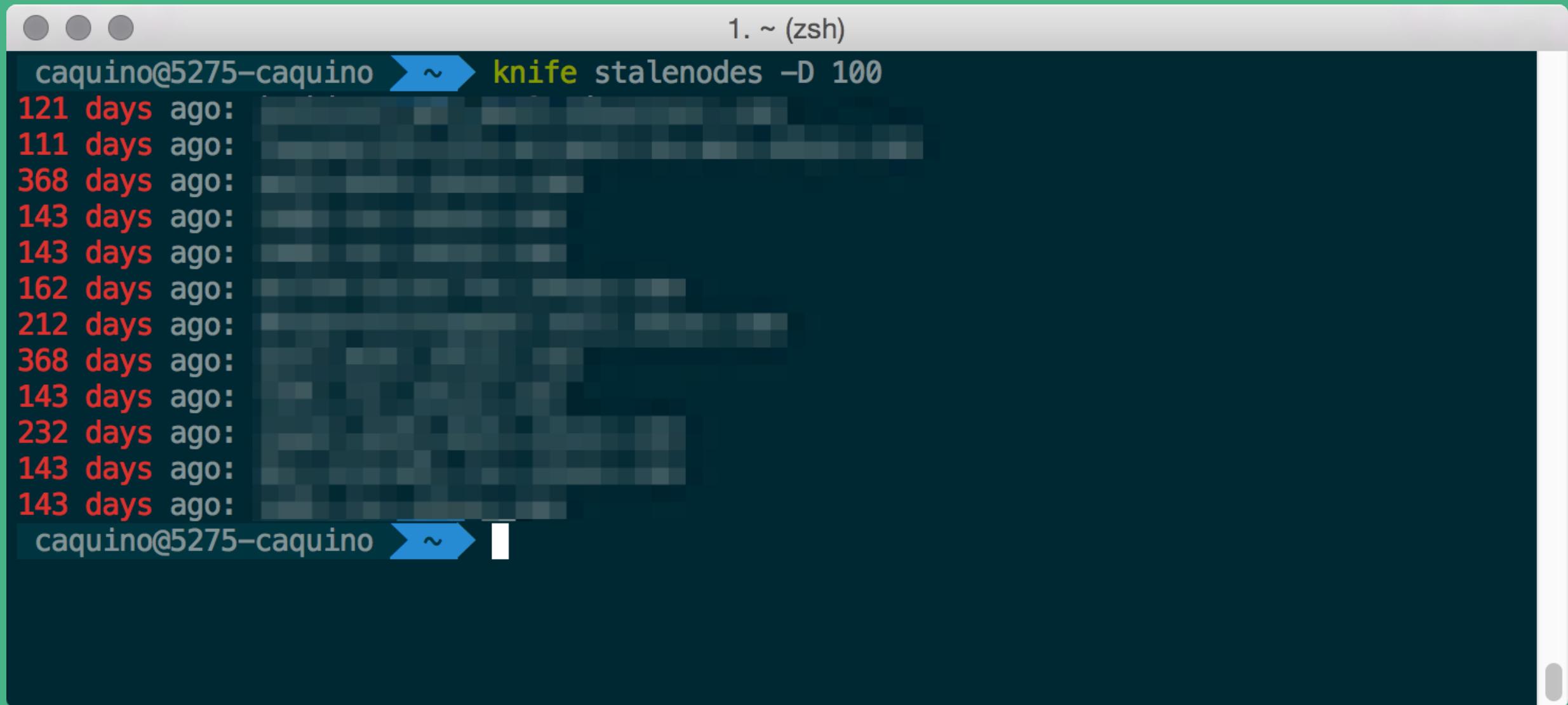
# PSEARCH

gem install knife-psearch

```
1. ~ (zsh)
caquino@5275-caquino ~ ➤ time knife search node 'fqdn:*' -i > /dev/null
knife search node 'fqdn:*' -i > /dev/null 228.15s user 21.00s system 22% cpu 18:24.04 total
caquino@5275-caquino ~ ➤ time knife psearch node 'fqdn:*' -i > /dev/null
knife psearch node 'fqdn:*' -i > /dev/null 4.49s user 1.65s system 28% cpu 21.783 total
caquino@5275-caquino ~ ➤ █
```

# STALE NODES

gem install knife-stalenodes

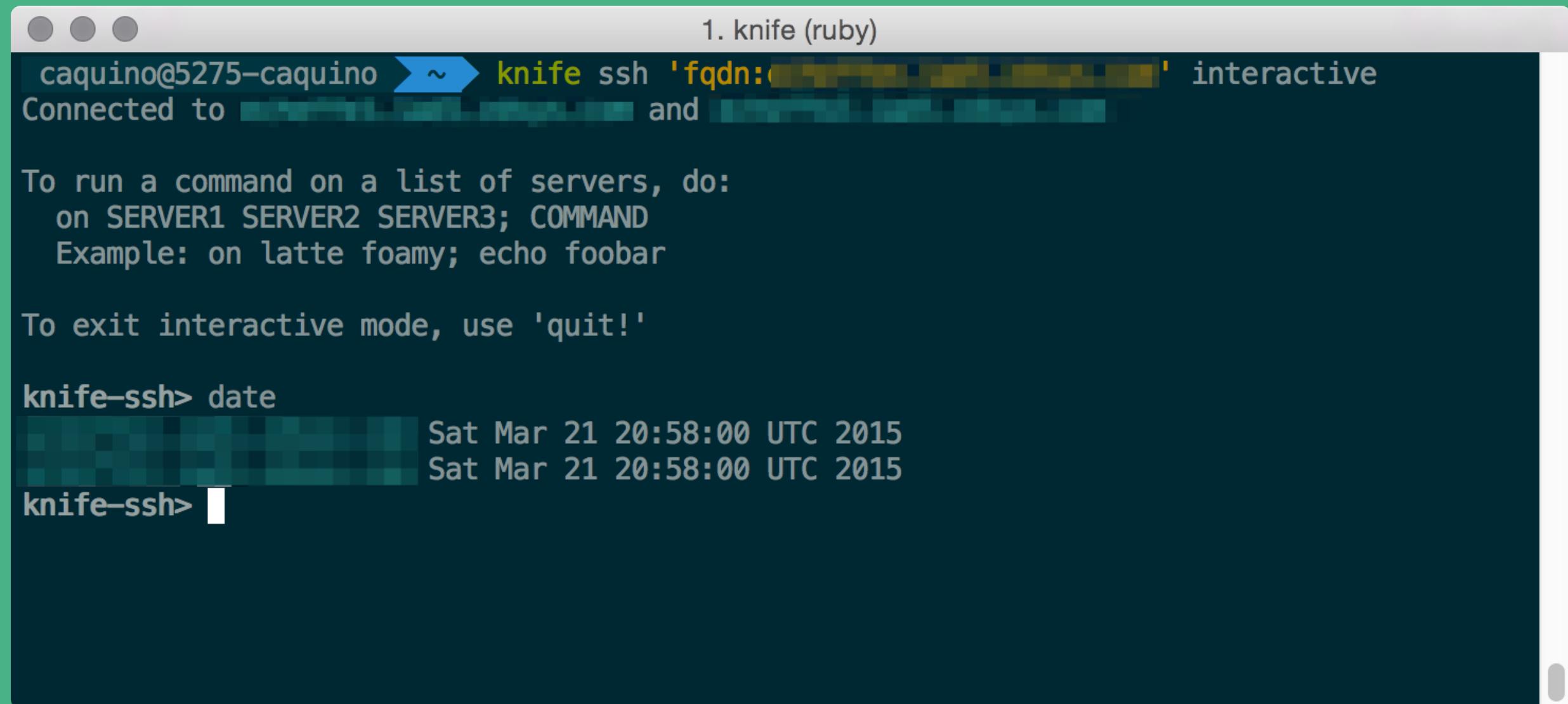


A screenshot of a terminal window titled "1. ~ (zsh)". The command "knife stalenodes -D 100" is run by the user "caquino@5275-caquino". The output lists several nodes that have been stale for a specific number of days, all of which are red text. The list includes:

- 121 days ago:
- 111 days ago:
- 368 days ago:
- 143 days ago:
- 143 days ago:
- 162 days ago:
- 212 days ago:
- 368 days ago:
- 143 days ago:
- 232 days ago:
- 143 days ago:
- 143 days ago:

# KNIFE SSH INTERACTIVE

knife ssh 'fqdn:\*.domain.com' interactive



The screenshot shows a terminal window titled "1. knife (ruby)". The command entered is "knife ssh 'fqdn:\*.domain.com' interactive". The output indicates that the user is connected to two servers. The terminal then provides instructions for running commands on multiple servers and shows examples. Finally, it prompts the user to exit interactive mode with "quit!". A command "date" is run, showing the same timestamp on both connected servers.

```
1. knife (ruby)
caquino@5275-caquino ~ ➤ knife ssh 'fqdn:*.domain.com' interactive
Connected to [REDACTED] and [REDACTED]

To run a command on a list of servers, do:
  on SERVER1 SERVER2 SERVER3; COMMAND
  Example: on latte foamy; echo foobar

To exit interactive mode, use 'quit!'

knife-ssh> date
[REDACTED] Sat Mar 21 20:58:00 UTC 2015
[REDACTED] Sat Mar 21 20:58:00 UTC 2015
knife-ssh> █
```

# SEARCH

All about attributes

- knife psearch node "(NOT kernel\_machine:x86\_64)"
- knife psearch node "virtualization\_role:guest"

Range

- knife psearch node "zendesk\_config\_pod:[4 TO 6]"

Get to know your attributes

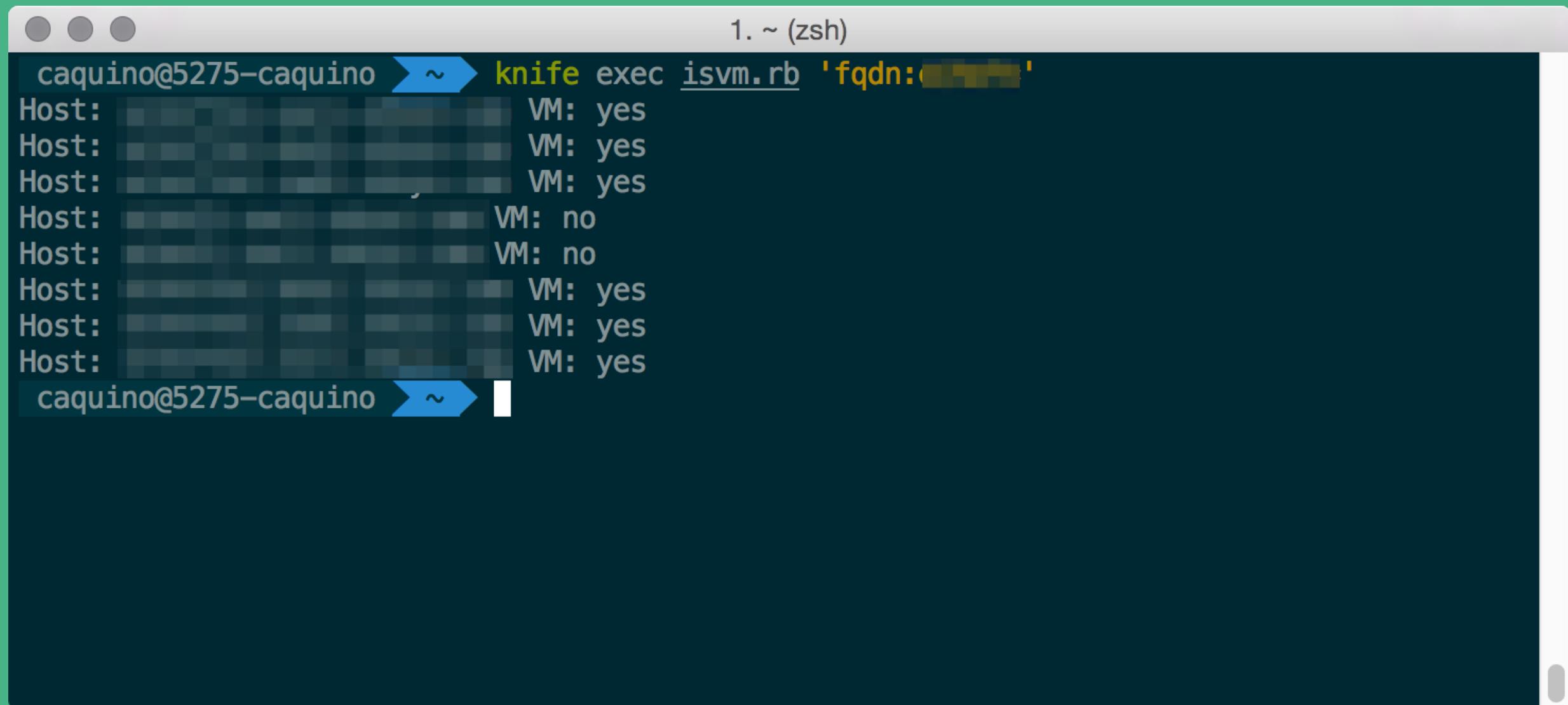
- knife node show -l node.domain.com

# KNIFE EXEC SCRIPT

```
Chef::Log.level = :fatal
search( :node, ARGV[2] ).each do |node|
  vm = (node["virtualization"] &&
         node["virtualization"]["role"] == "guest") ? 'yes' : 'no'
  puts "Host: #{node['fqdn']} VM: #{vm}"
end
exit 0
```

# KNIFE EXEC

```
knife exec is_vm.rb 'fqdn:*.mydomain.com'
```



A screenshot of a terminal window titled "1. ~ (zsh)". The command entered is "knife exec isvm.rb 'fqdn:\*.mydomain.com'". The output shows a list of hosts with their VM status:

Host	VM
Host 1	VM: yes
Host 2	VM: yes
Host 3	VM: yes
Host 4	VM: no
Host 5	VM: no
Host 6	VM: yes
Host 7	VM: yes
Host 8	VM: yes

QUESTIONS?  
THANKS

**MADE WITH  
ZENDESK**